<center>**REMARKS**</center>

Claims 1-8, 10-15, 17-23, and 27 are currently pending in the subject application and are presently under consideration. Claims 1, 14, 15, 23, and 27 have been amended. A listing of claims is shown at pages 2-8. Favorable reconsideration of the subject patent application is respectfully requested in view of the amendments and comments herein.

I.   Rejection of Claims 1-8 and 9-14 Under 35 U.S.C. §103(a)

Claims 1-8 and 9-14 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Arnold, *et al.* (US 6,393,497), in view of Hollander, *et al.* (US 6,823,460), in further view of Clarke, *et al.* (US 2002/0035642). Withdrawal of this rejection is respectfully requested in view of at least the following reasons. Arnold, *et al.* alone or in combination with Hollander, *et al.* and/or Clarke, *et al.* does not teach or suggest each and every aspect of the subject claims.

> *[T]he prior art reference (or references when combined) must teach or suggest all the claim limitations.* See MPEP § 706.02(j). *See also KSR Int'l Co. v. Teleflex, Inc.*, 550 U. S. ____, 04-1350, slip op. at 14 (2007). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must be found in the prior art and not based on applicant's disclosure. *See In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (emphasis added).

Applicant's subject claims relate to a system and method that facilitates application developers creating proxies, accessing method call interception functionality, retrieving information associated with a method call that can be intercepted by the interception functionality and adapting and/or extending the functionality of object systems. Specifically, independent claims 1 and 14 relate to similar aspects, namely, *the optimization of the remote method call includes achieving an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked.* Arnold, *et al.*, Hollander, *et al.* and Clarke, *et al.*, alone or in combination, fail to teach this novel aspect.

<center>9</center>

Arnold, *et al.* relates to a system and method that employs a smart proxy as a wrapper around a stub in a distributed system. The system transmits a request for a particular object and receives a response to the request that includes code used to construct a representation of the requested object, the construction creating an object for processing calls to the object, local to the requesting object, using the representation. However, the system disclosed by Arnold, *et al.* fails to teach a component for method call interception that can be accessed by application code to adapt and/or extend system functionalities of the system. Further, Arnold, *et al.* does not teach or suggest optimizing remote method calls by determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object and/or determining which objects are invoked, such that response time is optimized.

The Examiner asserts in the Advisory Action (dated October 10, 2008) that Hollander, *et al.* teaches optimizing the remote method call includes at least one of determining an amount of data sent to the object, determining a type of data sent to the object or determining which objects are invoked. However, Hollander, *et al.* merely relates to a method and system whereby API functions called by user applications are not allowed to execute unless the calling process has the requisite authority and privilege. Specifically, API function arguments are processed and decision is made by a Pre-Entry routine whether the API function is allowed to execute in the present environment. Such decision may rely on user predefined information or on online decision making by a user or a system administrator controlling operation of the operating systems. Such decision could further be made according to static pre-defined parameters or dynamically according to the type and value of the arguments passed by the calling applications. However, Hollander, *et al.* does not teach or suggest optimizing the remote method call to achieve an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked. In particular, the system disclosed by Hollander, *et al.* merely determines whether the API function is allowed to execute in the present environment based on the type and value of the arguments passed by the calling applications but fails to teach or suggest an application code generic proxy that performs proxy pre-processing comprising transaction processing and machine learning to

determine an amount of data sent to the object, determine a type of data sent to the object and/or determine which objects are invoked before invoking a method on the object as recited by the subject claims. The system disclosed by Hollander, *et al.* teaches identification of the type and value of the arguments passed by the calling applications to determine if the API function is allowed to execute in the present environment but does not teach or suggest determining an optimal type or value of the arguments and passing the optimal type/value to optimize (reduce) response time.

Applicants' subject claims, in contrast, relate to a system that employs application developer coded extensible proxies that have access to method interception and remote functionality and data, wherein remote method call invocation is optimized. The disclosed system facilitates method call interception wherein control passes from the method caller to a proxy, rather than from the method caller to a remote object. While such interception and routing is conventionally performed by system code, *the disclosed system facilitates gaining access to such interception and routing functionality.* Thus, the proxy includes a customized proxy component that can be written in application code by application programmers and which has access to the interception and routing functionality provided by system code. The customized proxy component can be, therefore, operable to adapt and/or extend the functionality provided by a conventional proxy. Specifically, application code may be employed in actions including, but not limited to, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation. (*See* page 11, lines 3-25.) By providing application programmers with the opportunity to produce and custom craft the customized proxy and thus to engage in proxy pre-processing, the proxy can be adapted and/or extended beyond the capabilities provided by conventional systems. Since the customized proxy had the opportunity to perform proxy pre-processing, *the invocation includes optimization of the amount of data sent to the remote object, the type of data sent to the remote object and which remote object(s), if any, are invoked.* According to an aspect, the proxy pre-processing may have determined that a database value had previously been cached in the proxy, and thus the database value could be passed to the remote object and thus the database lookup may not occur. For example, in the web browser/stock market data feed

example, a user may have previously inquired about a stock price for a company named AABBCC company. Before the stock price could be retrieved, the remote object may have had to look up a ticker symbol for the company (*e.g.,* AACo), and that ticker symbol may have been cached in the customized proxy. Thus, the ticker symbol, rather than the name of the company, may be sent to the remote object which can eliminate a duplicate lookup in the remote object. By way of further illustration, proxy pre-processing performed by the customized proxy may have determined that a second remote object may be better suited to handle the method invoked by the method caller than a first remote object, and thus the customized proxy may invoke the method on the second remote object instead of the first remote object. For example, returning to the browser/stock market data feed example, the customized proxy may have determined that recent stock market inquiries on a first remote object have taken sixty seconds, while a response time of five seconds is desired. *Thus, the customized proxy may acquire the stock market price from a different remote object and compare the average response times to determine how more optimal response times can be achieved.* Thus, increases in performance may be achieved for the method caller. The cited references are silent with respect to these novel aspects.

Clarke, *et al.* fails to make up for these deficiencies of Hollander, *et al.* and Arnold, *et al.* Clarke, *et al.* teaches a system and method that manages network traffic by employing an intermediary node, such as a proxy, that implements a flow control algorithm to avoid network congestion. The system includes a server that returns an error response when it receives a request that it cannot handle from the client. This response is passed back to the client *via* a proxy, which recognizes the response type and learns a back off time for the server. Incase a disparate client sends a request to the server, the proxy returns a back off signal to the disparate client and reduces the number of requests reaching the congested server. However, Clarke, *et al.* fails to disclose *at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked for optimizing a remote method call to optimize response times.*

In view of at least the foregoing, it is clear that Arnold, *et al.*, Hollander, *et al.* and Clarke, *et al.*, alone or in combination, do not teach or suggest applicants' claimed

subject matter as recited in independent claims 1 and 14 (and claims 2-8 and 9-13 which respectively depend there from) and thus fail to make obvious the subject claims. Thus, this rejection should be withdrawn.

II.   Rejection of Claims 15 and 20-22 Under 35 U.S.C. §103(a)

Claims 15 and 20-22 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Colyer (US 5,903,725), in view of Clarke, *et al.* (US 2002/0035642), in further view of Hollander, *et al.* (US 6,823,460). Withdrawal of this rejection is respectfully requested in view of at least the following reasons. The cited references fail to teach or suggest each and every limitation of the subject claims.

Applicants' claimed subject matter relates to a system for interacting with an object that includes a method call interceptor to intercept a method call to an object and route the method call to a proxy. Unlike conventional systems, the method call interceptor is accessible to application code. Further, the system also includes an application code generic proxy that can receive an intercepted method call, that can invoke the method on the object imaged by the proxy, that can receive results from the object imaged by the proxy and that can pass results to the entity that generated the intercepted method call. Unlike conventional systems, the proxy can be written by application developers. Specifically, independent claim 15, recites *wherein the optimizing the remote method call includes achieving an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked.* Additionally, independent claim 15 recites, *performing custom, user-directed application processing to at least one of monitor or control the processing of a message between the application code generic proxy and the object.* Colyer, Clarke, *et al.* and Hollander, *et al.*, alone or in combination, fail to teach or suggest these novel aspects.

Colyer teaches a system that creates recoverable proxies to protect a server against invalid usage of proxy objects. In particular, the system transparently re-creates proxy objects in a client of a client-server distributed processing system. On malfunction of a server, and consequently invalidity of the proxy objects, a proxy register object causes all proxy objects to be refreshed. However, Colyer fails to disclose interception of

a method call that is made accessible to a developer to at least one of adapt or extend functionalities of a proxy and does not teach or suggest determination of an amount of data sent to the object, a type of data sent to the object or determination which objects are invoked. Further, Colyer is silent with respect to monitoring or controlling the processing of a message between the application code generic proxy and the object. As discussed *supra*, Clarke, *et al.* and Hollander, *et al.*, do not disclose *the optimizing the remote method call includes achieving an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked* and thus fail to cure the aforementioned deficiencies of Colyer with respect to independent claim 15. Accordingly, withdrawal of this rejection is requested.

III.   Rejection of Claims 17-19, 23, and 27 Under 35 U.S.C. §103(a)

Claims 17-19, 23, and 27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Colyer (US 5,903,725), Clarke *et al.* (US 2002/0035642), and Hollander *et al.* (US 6,823,460), in further view of Arnold *et al.* (US 6,393,497). Withdrawal of this rejection is respectfully requested in view of at least the following reasons. The cited references, alone or in combination, fail to teach or suggest each and every limitation of applicants' claimed invention.

Independent claims 15, 23 and 27 relate *to optimization of a remote method call to achieve an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked*. As discussed above, Colyer and Clarke, *et al.*, alone or in combination do not disclose a system or method that intercepts a method call and optimizes the method call by determining an optimal amount of data sent to the object, determining an optimal type of data sent to the object or determining which objects are invoked, such that response time is optimized. Furthermore, Arnold, *et al.* and Hollander, *et al.* are silent with respect to a method call interceptor that is accessible to application code for adapting and/or extending system functionalities of a proxy and fail to remedy the aforementioned deficiencies of Colyer and Clarke, *et al.* Accordingly, it is respectfully requested that this rejection be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP243US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
127 Public Square
57th Floor, Key Tower
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731